In the Claims:

Please amend claims 1, 6, 8, 14, 19, 21, 27, 28, 29, 30, 31, 33, 34, and 35, as indicated below.

1. (Currently amended) A microprocessor, comprising:

an instruction cache configured to store instructions;

a branch prediction unit;

a trace cache configured to store a plurality of traces of instructions; and

a prefetch unit coupled to the instruction cache, the branch prediction unit, and the trace cache;

wherein the prefetch unit is configured to fetch instructions from the instruction cache until the branch prediction unit outputs a predicted target address;

wherein the prefetch unit is configured to check the trace cache for a match for the predicted target address in response to the branch prediction unit outputting [[a]] the predicted target address;

wherein the prefetch unit is configured to not check the trace cache for a match until the branch prediction unit outputs [[a]] the predicted target address; and

wherein [[if]] in response to the prefetch unit ~~identifies~~ identifying a match for the predicted target address in the trace cache, the prefetch unit is configured to fetch one or more of the plurality of traces from the trace cache.

2. (Original) The microprocessor of claim 1, wherein the branch prediction unit is configured to output the predicted target address in response to a prediction that a branch will be taken.

3. (Original) The microprocessor of claim 1, wherein the branch prediction unit is configured to output the predicted target address in response to detection of a branch misprediction.

4. (Original) The microprocessor of claim 1, further comprising a trace generator, wherein the trace generator is configured to begin a trace with an instruction corresponding to a label boundary.

5. (Original) The microprocessor of claim 4, wherein the trace generator is configured to check the trace cache for a duplicate copy of the trace that the trace generator is constructing.

6. (Currently amended) The microprocessor of claim 5, wherein [[if]] in response to the trace generator ~~identifies~~ identifying a duplicate copy of the trace, the trace generator is configured to discard the trace under construction.

7. (Previously presented) The microprocessor of claim 5, wherein in response to the trace generator identifying an entry corresponding to a duplicate copy of the trace, the trace generator is configured to check the trace cache for an entry corresponding to a next trace to be generated.

8. (Currently amended) The microprocessor of claim 7, wherein [[if]] in response to the trace generator ~~identifies~~ identifying a trace entry corresponding to the next trace to be generated, the trace generator is configured to discard the trace under construction.

9. (Original) The microprocessor of claim 4, wherein the trace generator is configured to generate traces in response to instructions being retired.

10. (Original) The microprocessor of claim 4, wherein the trace generator is configured to generate traces in response to instructions being decoded.

11. (Original) The microprocessor of claim 1, wherein each of the plurality of traces comprises partially-decoded instructions.

12. (Original) The microprocessor of claim 1, wherein each of the plurality of traces is associated with a tag comprising the address of an earliest instruction, in program order, stored within that trace.

13. (Original) The microprocessor of claim 1, wherein each of the plurality of traces is associated with a flow control field comprising a label for an instruction to which control will pass for each branch operation comprised in that trace.

14. (Currently amended) A computer system, comprising:

a system memory; and

a microprocessor coupled to the system memory, comprising:

an instruction cache configured to store instructions;

a branch prediction unit;

a trace cache configured to store a plurality of traces of instructions; and

a prefetch unit coupled to the instruction cache, the branch prediction unit, and the trace cache;

wherein the prefetch unit is configured to fetch instructions from the instruction cache until the branch prediction unit outputs a predicted target address;

wherein the prefetch unit is configured to check the trace cache for a match for the predicted target address in response to the branch prediction unit outputting the predicted target address;

wherein the prefetch unit is configured to not check the trace cache for a match until the branch prediction unit outputs [[a]] the predicted target address; and

wherein [[if]] in response to the prefetch unit ~~identifies~~ identifying a match for the predicted target address in the trace cache, the prefetch unit is configured to fetch one or more of the plurality of traces from the trace cache.

15. (Original) The computer system of claim 14, wherein the branch prediction unit is configured to output the predicted target address in response to a prediction that a branch will be taken.

16. (Original) The computer system of claim 14, wherein the branch prediction unit is configured to output the predicted target address in response to detection of a branch misprediction.

17. (Original) The computer system of claim 14, further comprising a trace generator, wherein the trace generator is configured to begin a trace with an instruction corresponding to a label boundary.

18. (Original) The computer system of claim 17, wherein the trace generator is configured to check the trace cache for a duplicate copy of the trace that the trace generator is constructing.

19. (Currently amended) The computer system of claim 18, wherein [[if]] in response to the trace generator ~~identifies~~ identifying a duplicate copy of the trace, the trace generator is configured to discard the trace under construction.

20. (Previously presented) The computer system of claim 18, wherein in response to the trace generator identifying an entry corresponding to a duplicate copy of the trace, the trace generator is configured to check the trace cache for an entry corresponding to a next trace to be generated.

21. (Currently amended) The computer system of claim 20, wherein [[if]] in response to the trace generator ~~identifies~~ identifying a trace entry corresponding to the next trace to be generated, the trace generator is configured to discard the trace under construction.

22. (Original) The computer system of claim 17, wherein the trace generator is configured to generate traces in response to instructions being retired.

23. (Original) The computer system of claim 17, wherein the trace generator is configured to generate traces in response to instructions being decoded.

24. (Original) The computer system of claim 14, wherein each of the plurality of traces comprises partially-decoded instructions.

25. (Original) The computer system of claim 14, wherein each of the plurality of traces is associated with a tag comprising the address of an earliest instruction, in program order, stored within that trace.

26. (Original) The computer system of claim 14, wherein each of the plurality of traces is associated with a flow control field comprising a label for an instruction to which control will pass for each branch operation comprised in that trace.

27. (Currently amended) A method, comprising:

receiving a retired instruction;

~~starting construction of a new trace if the received instruction is associated with a branch label;~~

determining if a previous trace under construction duplicates a trace in a trace cache and if the received instruction corresponds to a branch label;[[,]] and

in response to determining that a previous trace under construction duplicates a trace in a trace cache and that the received instruction corresponds to a branch label, ~~delaying~~ beginning construction of [[the]] a new trace ~~until the received instruction corresponds to a branch label~~.

28. (Currently amended) The method of claim 27, further comprising continuing construction of an incomplete trace already in process in response to determining that the incomplete trace does not duplicate a trace in a trace cache.

29. (Currently amended) The method of claim 27, further comprising searching the trace cache for duplicate entries subsequent to completion of the previous trace under construction or the new trace.

30. (Currently amended) The method of claim 29, further comprising creating a new entry in the trace cache [[if]] in response to no duplicate entry [[is]] being identified.

31. (Currently amended) The method of claim 29, further comprising discarding a trace [[if]] in response to a duplicate entry [[is]] being identified.

32. (Previously presented) A method, comprising:

fetching instructions from an instruction cache;

continuing to fetch instructions from the instruction cache without searching a trace cache until a branch target address is generated;

in response to a branch target address being generated, searching a trace cache for an entry corresponding to the branch target address.

33. (Currently amended) The method of claim 32, further comprising continuing to fetch instructions from the instruction cache [[if]] in response to no entry [[is]] being identified in the trace cache corresponding to the branch target address.

34. (Currently amended) The method of claim 32, further comprising fetching one or more traces from the trace cache [[if]] in response to an entry [[is]] being identified in the trace cache corresponding to the branch target address.

35. (Currently amended) A microprocessor, comprising:

means for receiving a retired operation;

~~means for starting a new trace if the received operation is a first operation at a branch label;~~

means for determining if a previous trace under construction duplicates a trace in a trace cache and if the received operation is a first operation at a branch label; and

means for ~~delaying~~ starting a new trace [[if]] in response to determining that a previous trace under construction duplicates a trace in a trace cache and that the received operation is a first operation at a branch label~~, until the received operation corresponds to a branch label~~.